Stats 101C: Homework 3

Kitu Komya (404491375)

May 3, 2017

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.3
```

```
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 3.3.3
```

```
library(boot)
library(resample)
```

```
## Warning: package 'resample' was built under R version 3.3.2
```

1a.

```
dim(Carseats)
```

```
## [1] 400  11
```

```
summary(Carseats)
```

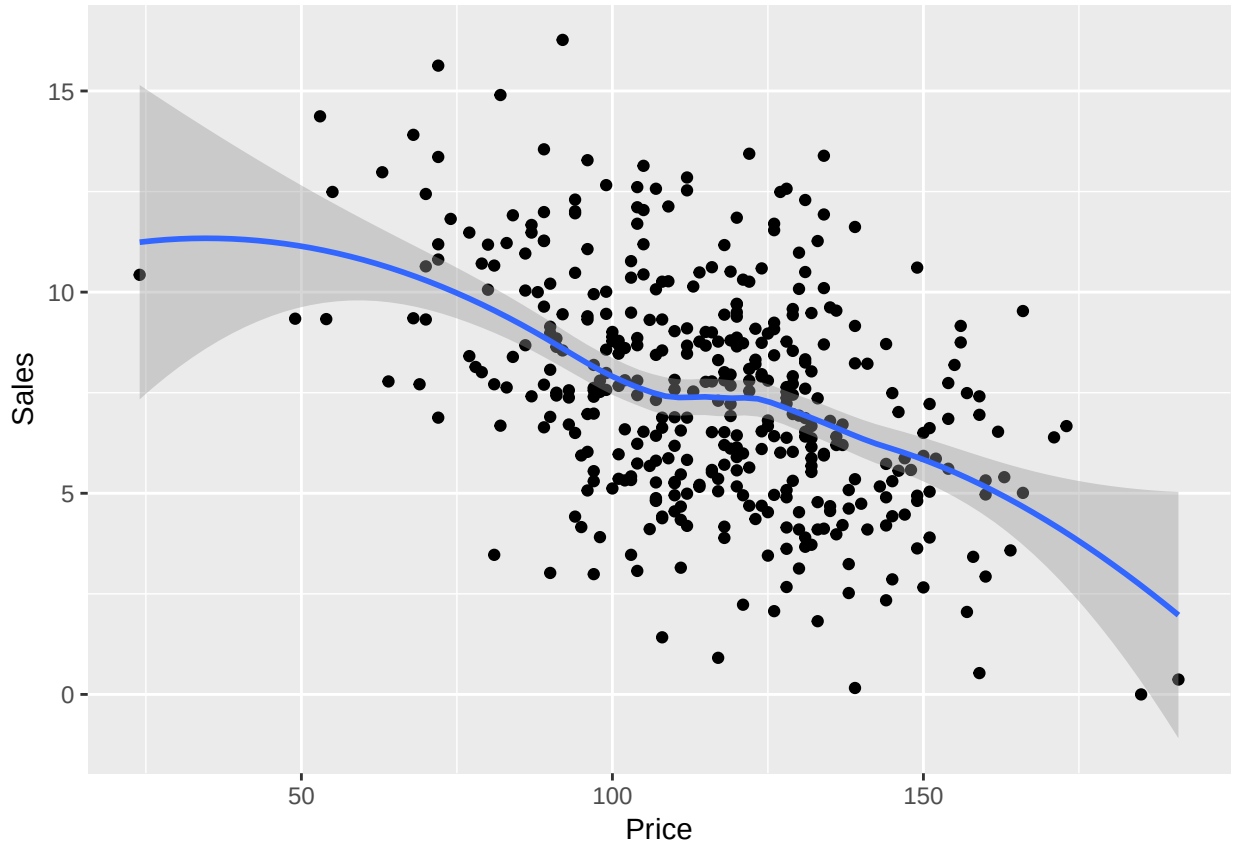
```
##      Sales          CompPrice       Income        Advertising
##  Min.   : 0.000   Min.   : 77   Min.   : 21.00   Min.   : 0.000
##  1st Qu.: 5.390   1st Qu.:115   1st Qu.: 42.75   1st Qu.: 0.000
##  Median : 7.490   Median :125   Median : 69.00   Median : 5.000
##  Mean   : 7.496   Mean   :125   Mean   : 68.66   Mean   : 6.635
##  3rd Qu.: 9.320   3rd Qu.:135   3rd Qu.: 91.00   3rd Qu.:12.000
##  Max.   :16.270   Max.   :175   Max.   :120.00   Max.   :29.000
##    Population        Price        ShelveLoc        Age
##  Min.   : 10.0   Min.   : 24.0   Bad   : 96   Min.   :25.00
##  1st Qu.:139.0   1st Qu.:100.0   Good  : 85   1st Qu.:39.75
##  Median :272.0   Median :117.0   Medium:219   Median :54.50
##  Mean   :264.8   Mean   :115.8                Mean   :53.32
##  3rd Qu.:398.5   3rd Qu.:131.0                3rd Qu.:66.00
##  Max.   :509.0   Max.   :191.0                Max.   :80.00
##    Education    Urban       US
##  Min.   :10.0   No :118   No :142
##  1st Qu.:12.0   Yes:282   Yes:258
##  Median :14.0
##  Mean   :13.9
##  3rd Qu.:16.0
##  Max.   :18.0
```

The dimension of this data frame is 400 by 11. The summary statistics for each variable are shown.

1b.

```
qplot(x = Price, y = Sales, geom = c("point", "smooth"), data = Carseats)
```

```
## `geom_smooth()` using method = 'loess'
```

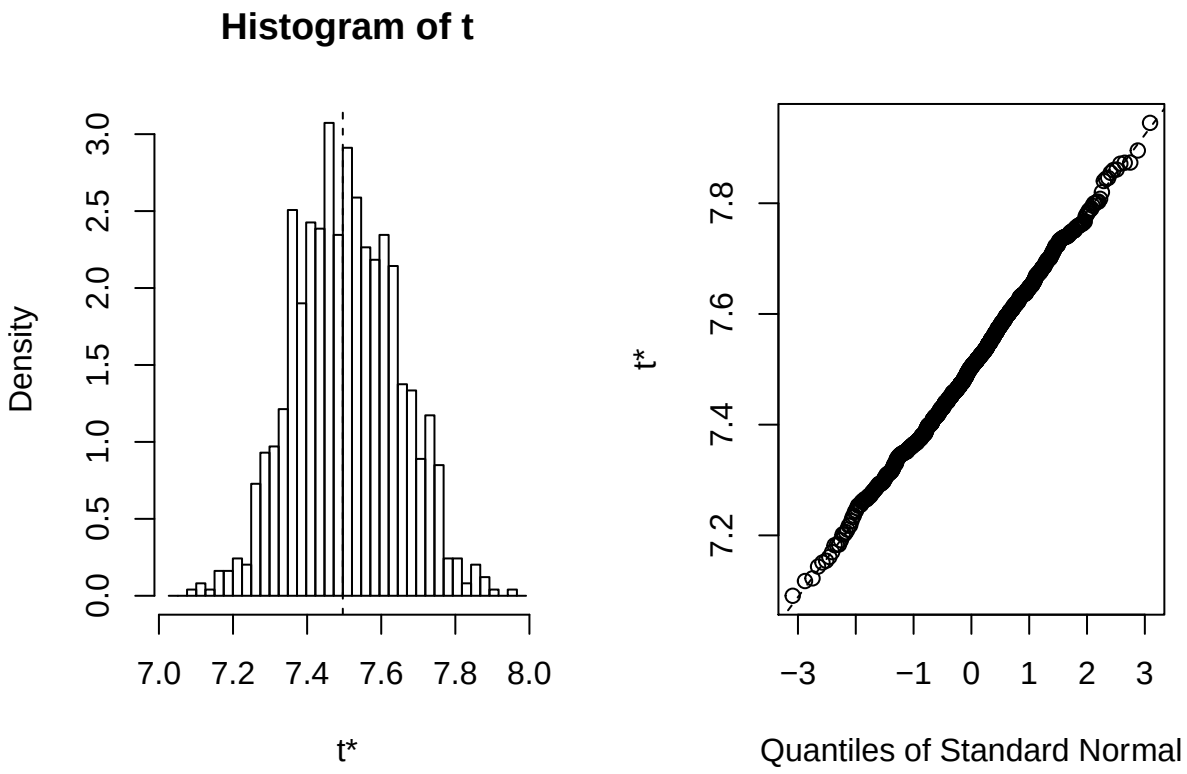


The graph shows that although there is a negative, moderately strong linear relationship, there is a lot of variability at the extremes, which is probably due to their being fewer points at the extremas, as the majority of the data is clustered near the center.

1c. & 1d.

```
# mean of sales using boot package
my_mean <- function(data, indices)
{
  return(mean(data[indices]))
}

# median of sales using boot package
my_median <- function(data, indices)
{
  return(median(data[indices]))
}
```

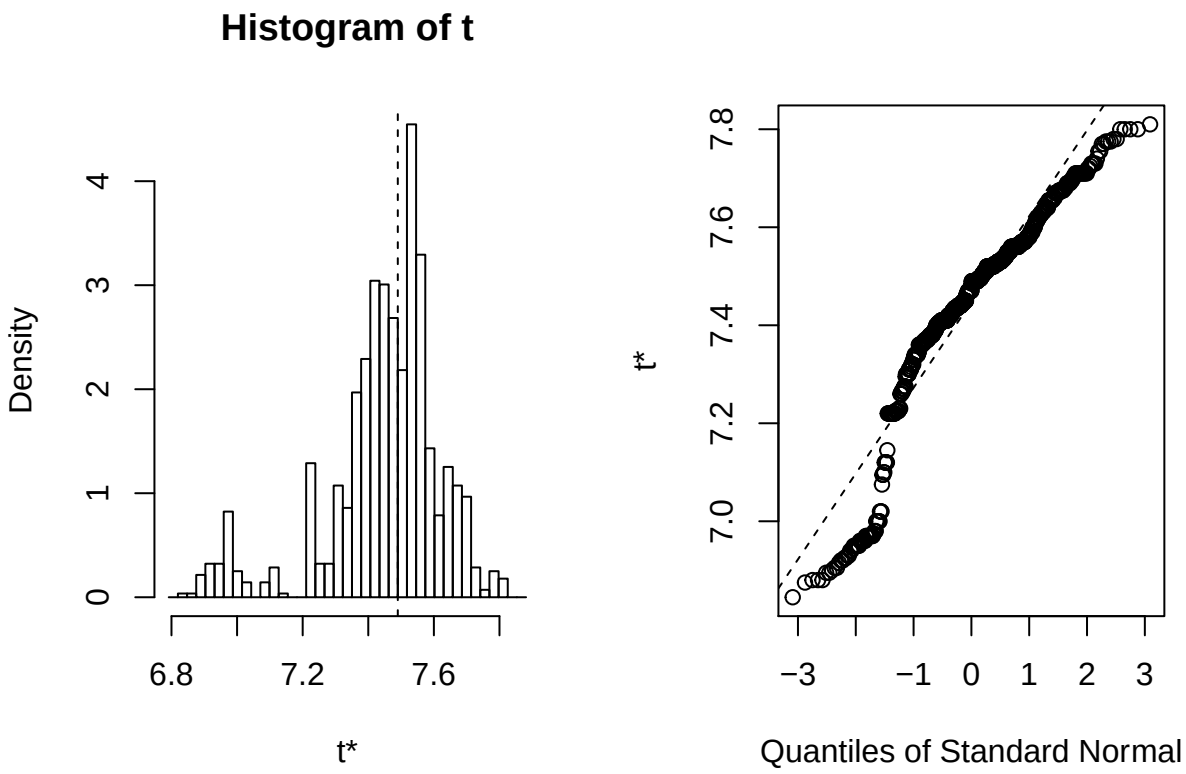
```
out_bs_mean <- boot(data = Carseats$Sales, statistic = my_mean, R = 1000)
out_bs_median <- boot(data = Carseats$Sales, statistic = my_median, R = 1000)

se_sales_mean <- sd(out_bs_mean$t)
se_sales_median <- sd(out_bs_median$t)

# confidence interval and plot mean, normal
plot(out_bs_mean)
```




```
c(mean(Carseats$Sales)-1.96*se_sales_mean, mean(Carseats$Sales) + 1.96*se_sales_mean)
```

```
## [1] 7.223467 7.769183
```

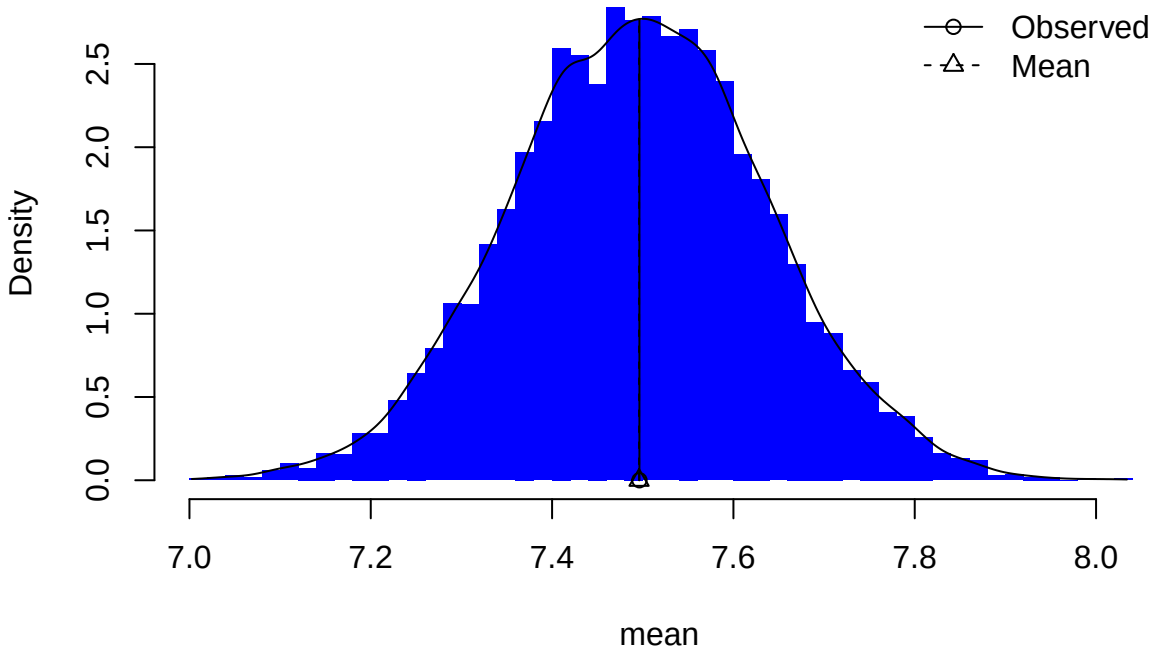
```
# median, skew
plot(out_bs_median)
```




```
low <- order(out_bs_median$t)[25]
high <- order(out_bs_median$t)[975]
c(out_bs_median$t[low], out_bs_median$t[high])
```

```
## [1] 6.95 7.71
```

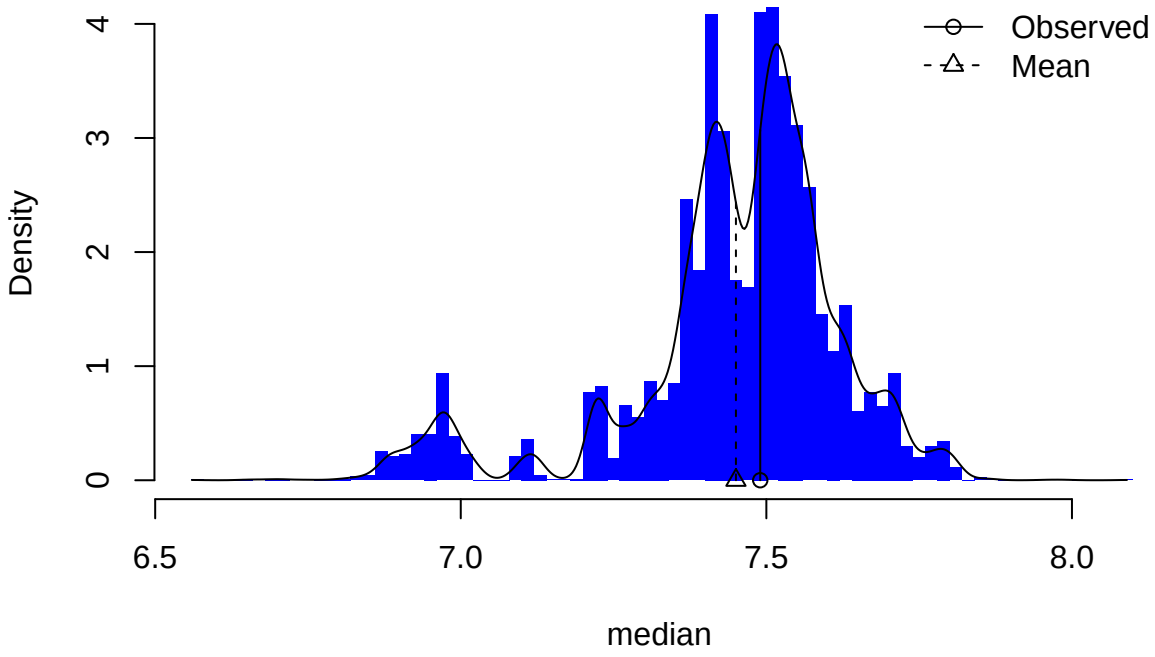
```
# boot package for mean and median of sales
out_bs_mean <- bootstrap(Carseats$Sales, mean)
out_bs_median <- bootstrap(Carseats$Sales, median)
se_sales_mean <- sd(out_bs_mean$replicates)
se_sales_median <- sd(out_bs_median$replicates)

# confidence interval and plot mean, normal
plot(out_bs_mean)
```




```
c(mean(Carseats$Sales)-1.96*se_sales_mean, mean(Carseats$Sales) + 1.96*se_sales_mean)
```

```
## [1] 7.219984 7.772666
```

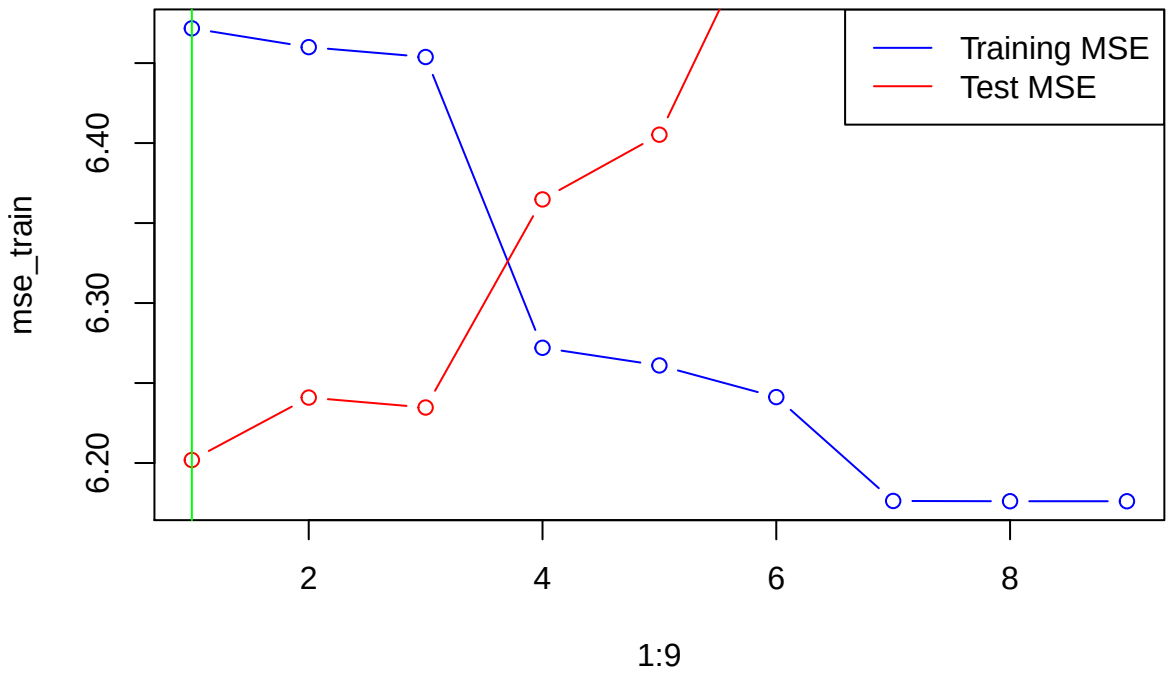
```
# median, skew
plot(out_bs_median)
```




```
#low2 <- order(out_bs_median$t)[25]
#high2 <- order(out_bs_median$t)[975]
#c(out_bs_median$t[low2], out_bs_median$t[high2])
```

2.

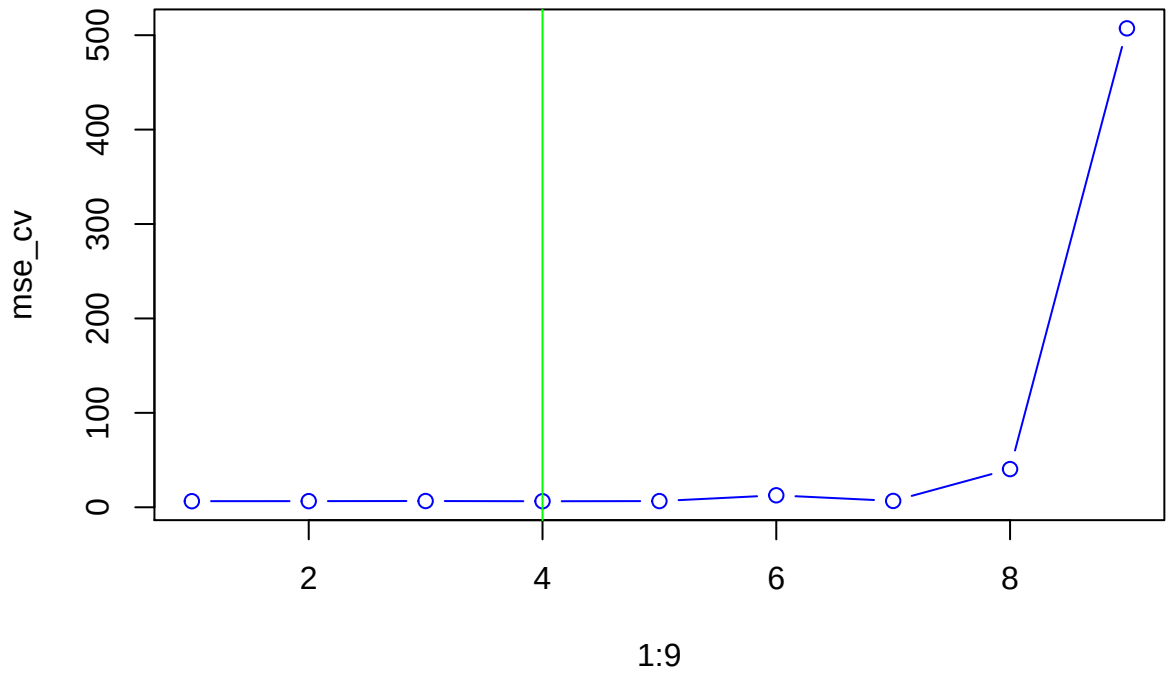
```
set.seed(77)
train <- sample(400, 280)

mse_train <- vector()
mse_test <- vector()

for (i in 1:9)
{
  lm_fit <- lm(Sales~poly(Price, i), data = Carseats, subset = train)
  mse_train[i] <- mean((Carseats$Sales-predict(lm_fit, Carseats))[train]^2)
  mse_test[i] <- mean((Carseats$Sales-predict(lm_fit, Carseats))[-train]^2)
}

# plots for part A and part B
plot(x = 1:9, y = mse_train, type = "b", col = "blue")
points(1:9, mse_test, type = "b", col = "red")
legend("topright", c("Training MSE", "Test MSE"), lty = c(1, 1), col = c("blue", "red"))
abline(v = which.min(mse_test), col = "green")
```



The best polynomial degree is 1 (linear) since the test MSE is the lowest at that point, as the green line also shows.

3.

```
set.seed(77)
mse_cv = vector()

for (i in 1:9)
{
  glm_fit <- glm(Sales~poly(Price, i), data = Carseats)
  mse_cv[i] <- cv.glm(Carseats, glm_fit)$delta[1]
}

plot(1:9, mse_cv, type = "b", col = "blue")
abline(v = which.min(mse_cv), col = "green")
```



The lowest MSE is at polynomial 4 as shown from the graph, so we would use polynomial 4.

4a.

```
set.seed(77)

# split into 10 folds
a <- split(sample(1:400), f = rep(1:10, 40))
a1 <- a[[1]]
a2 <- a[[2]]

head(Carseats[a1, ])
```

```
##     Sales CompPrice Income Advertising Population Price ShelveLoc Age
## 117  5.08       135     75           0        202   128    Medium  80
## 341  7.50       140     29           0        105    91       Bad  43
## 1    9.50       138     73          11        276   120       Bad  42
## 268  5.83       134     82           7        473   112       Bad  51
## 144  0.53       122     88           7         36   159       Bad  28
## 194 13.28       139     70           7         71    96      Good  61
##     Education Urban  US
## 117        10    No  No
## 341        16   Yes  No
## 1          17   Yes Yes
```

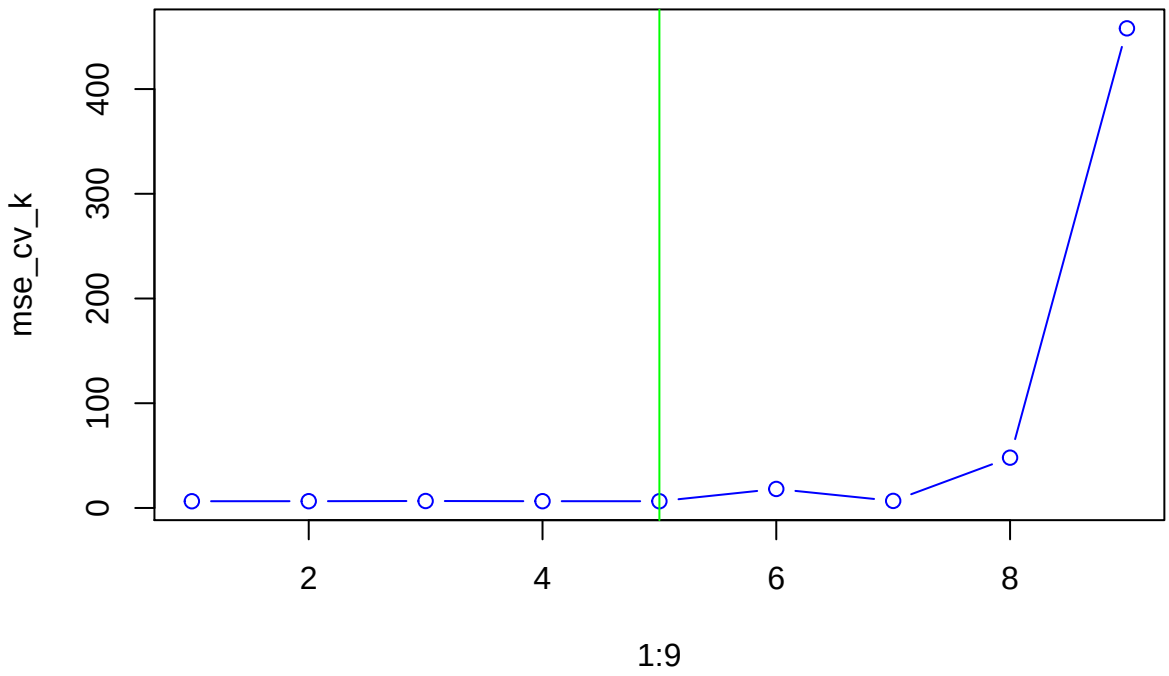
```
## 268        12    No Yes
## 144        17   Yes Yes
## 194        10   Yes Yes
```

```
head(Carseats[a2, ])
```

```
##     Sales CompPrice Income Advertising Population Price ShelveLoc Age
## 287  7.53       117    118          11        429   113    Medium  67
## 385 12.85       123     37          15        348   112      Good  28
## 393  4.53       129     42          13        315   130       Bad  34
## 232  8.09       132     69           0        123   122    Medium  27
## 59   5.42       103     93          15        188   103       Bad  74
## 394  5.57       109     51          10         26   120    Medium  30
##     Education Urban  US
## 287        18    No Yes
## 385        12   Yes Yes
## 393        13   Yes Yes
## 232        11    No  No
## 59         16   Yes Yes
## 394        17    No Yes
```

4b.

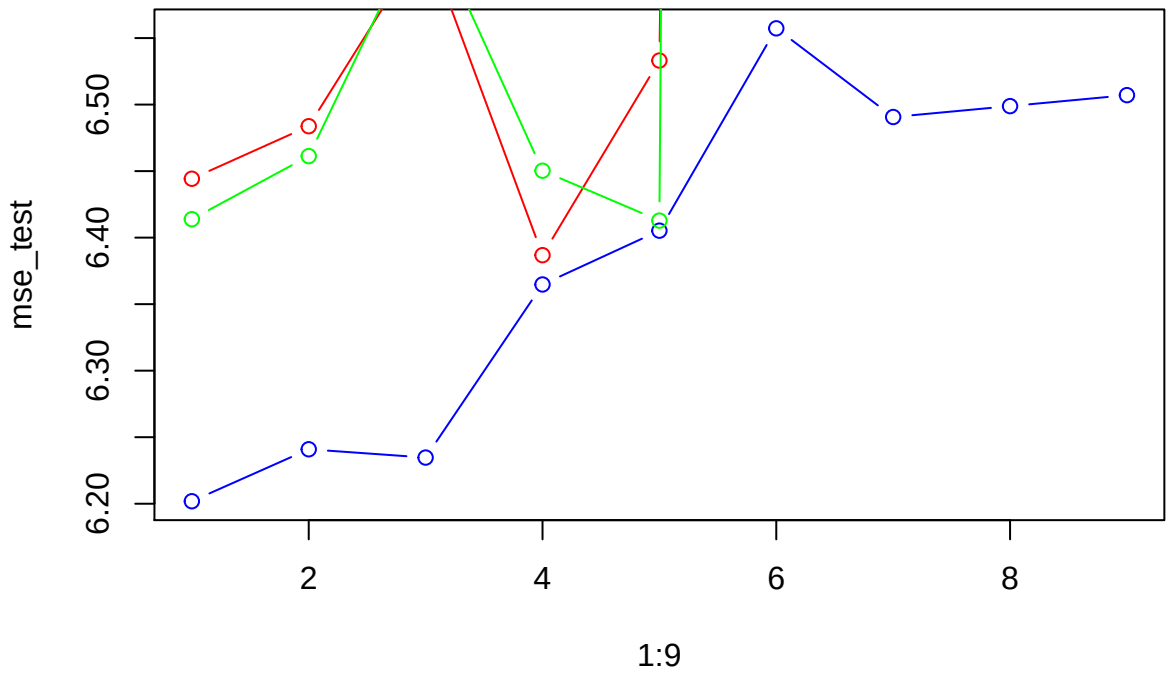
```
# using cross-validation technique
mse_cv_k <- vector()

for(i in 1:9)
{
  glm_fit <- glm(Sales~poly(Price, i), data = Carseats)
  mse_cv_k[i] <- cv.glm(Carseats, glm_fit, K = 10)$delta[1]
}

plot(1:9, mse_cv_k, type = "b", col = "blue")
abline(v = which.min(mse_cv_k), col = "green")
```



The best polynomial order is 5 since MSE is minimized here.

5.

```
plot(1:9, mse_test, type = "b", col = "blue")
points(1:9, mse_cv, type = "b", col = "red")
points(1:9, mse_cv_k, type = "b", col = "green")
```



We notice that at polynomial degree 4, the MSEs are minimized among all the lines. From this we conclude that 4 is the best polynomial for the model. Also note that the 10 fold method and LOOCV have very similar MSEs.